LAB 2
Sabawun Afzal Khattak 2328284 Ihab Ahmad 2328466 Hassaan Ali 2344810

OBJECTIVE: The purpose of the second laboratory module is the development of a more complex embedded system with advanced assembly programming and use of data structures like stacks.

2.4.1

b)

- ROM: Non-volatile memory that retains information in the event of a power failure. reading Only memory configured by a manufacturer that cannot be modified. Includes programming to boot electronic devices and performs some important tasks Contains a description of the program or software.

- SRAM: Static random-access memory, called SRAM, helps you get data faster. It is expensive because there are so many latches to store all the bits. There is no need to update the SRAM. During power supply, data is retained in memory and SRAM is mainly Caches computer memory and has low packing density

- DRAM: Dynamic random-access memory is known as DRAM, which is a type of random access. Memory commonly used by computers, workstations, and servers. DRAM stores each Data bit of another capacitor. It is a volatile memory because all data is lost when the power is cut off. road. RAM helps you access your data faster than storage media. Compared to SRAM It will slow down. Also, because DRAM uses capacitors, it consumes less power.

- SDRAM: Synchronous dynamic random-access memory is called SDRAM. SDRAM synchronizes the memory speed with the CPU clock speed. Thanks to SDRAM The SDRAM controller can react to the exact clock cycle during data preparation, feed to CPU and follow other instructions at known times.

- DDR3 SDRAM: Double data rate 3 SDRAM is called DDR3 SDRAM. As the name implies, it is synchronous dynamic random-access memory, Since the data rate is doubled, the transfer speed is higher than that of SDRAM. Similar use with SDRAM.

- Flash: Flash is non-volatile memory and is called flash memory or flash memory. Is often used for storage and data transfer between devices. Applications known as USB Flash drive, SSD, MP3 player, etc. Rewrite data at byte level. It uses a transistor Memory for storing single bits. It helps reduce costs.

- 6116 is a static ram.

c) It is required due to the ALE signal which is connected to the latch. The connection lets us change between lower and upper addresses. The latch acts as a communicator between the data bus of sram as well as the address bus to the MCU. Without this, would not be able to communicate.

2.4.1

(a) (i) 0x00

```
110101 | 00000000
         110101
         ------
         110101
         110101
         -------
         0000000   remainder.
```

No CRC error.
Reset Request.

(ii) 0x5f

```
110101 | 01011111
          110101  ↓
          0110101
           110101
          -------
           000000   remainder
```

No CRC error
Acknowledgement.

(iii) 0x6B

```
110101 | 01101011
          110101
         --------
         0000001   remainder.
```

CRC ERROR.

(iv) 0xA6 followed by 0x25

```
110101 | 1010 0110 0010 0101
         110101
        ---------
         0101001
          110101
        ---------
         00110000
           110101
          --------
             101010
             110101
            --------
              11110101
              110101
             ---------
                1011101
                110101
               --------
                 110111
                 110101
```

data packet followed by command packet.

→ 00010 remainder.

CRC error

(v) 0xF2 followed by 0x26

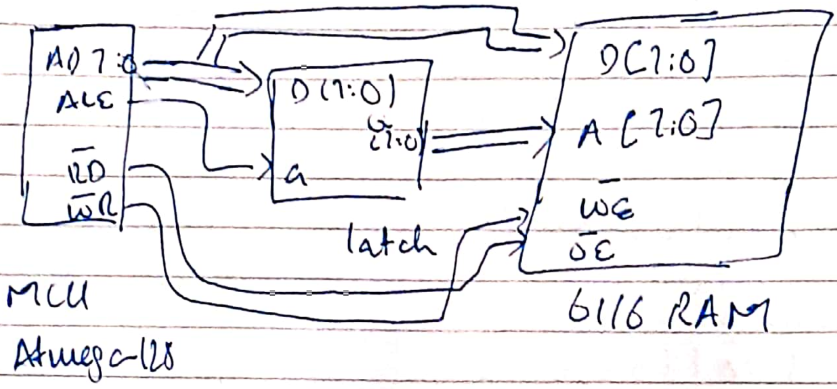
```
1111001000100110
110101
  10011000100110
  110101
   1001100100110
   110101
    100110100110
    110101
     10011100110
     110101
      1001000110
      110101
       100010110
       110101
        10111110
        110101
         1101010
         110101
          000000
          000000
          000000
```

Battery level data packet followed by command packet.

MO ERROR R=0.

2.4.2

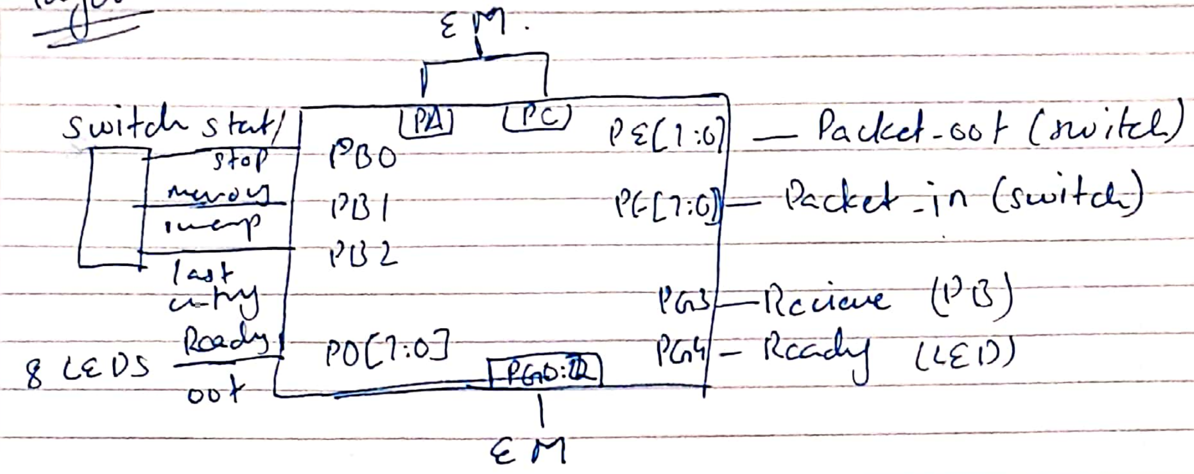
Address Decoding not required.



Inputs/outputs

PORT	USE
PORT A, C, G[0-2]	External Memory Connections.
PORTB	0 : Start / stop 1 : Memory Dump 2 : Last Entry Inputs
PORTD	Readout [7:0] output
PORTE	Packet.Out [7:0] output
PORTF	(Packet. In [7:0] (input)
PORT G	3 : Recieve (input) 4 : Ready (output).

layout


EM.
PA
PC
switch start/
stop
memory
inemp
last
entry
Ready
8 LEDS
out
PB0
PB1
PB2
PD[7:0]
PG0:2
EM
PE[7:0] — Packet-out (switch)
PF[7:0] — Packet_in (switch)
PG3 — Recieve (PB)
PG4 — Ready (LED)

Code:

```
; Note :: R21 HOLDS PACKET, TOS R23, R22 is empty



.include "m128def.inc"


.EQU Ones = 0xFF
.EQU Zeros= 0x00
.EQU Packet_Out = porte
.EQU Packet_In = pinf
.EQU Control_Switch = pinb
.EQU Read_Out = portd
.EQU Ready = portg
.EQU Receive = ping


.macro Port_Config

LDI R16, Zeros
OUT DDRB, R16 ; portb[0-2] input
STS DDRF, R16 ; portf for packet_in


LDI R16, Ones
OUT DDRD, R16 ; portd Read_out [8 LEDS]
OUT DDRE, R16 ; porte Packet_out
```

```
LDS R16, 0x10 ;
STS DDRG, R16 ; G(Ready[4], Recieve[3])


.endmacro


.macro XMEM_Config

IN R16, MCUCR
SBR R16, 128
OUT MCUCR, R16


.endmacro


.macro Pointer_Config


LDI xl, low(SRAM_START)
LDI xh, high(SRAM_START)


.endmacro


.macro Stack_Init


LDI R16, low(RAMEND)
```

```
OUT SPL, R16

LDI R16, high(RAMEND)

OUT SPH, R16



.endmacro

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;



.cseg

.org 0x0000

rjmp power



.org 0x0100

;;;;;;;;;;;;;;;;;;;;;;;;;SUBROUTINES/FUNCTIONS THAT THE PROGRAM
USES;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

crc11_execute:

LDI R18, 0b10110101

LDI R19, 0;

EOR R17,R18



crc11_execute_Loop:

CPI R19, 16-5;

BRSH crc11_end;



MOV R20, R17

ANDI R20, 0b10000000

BREQ crc11_execute_l1
```

```
EOR R17,R18
RJMP crc11_execute_Loop


crc11_execute_l1:
INC R19
ROL R16
ROL R17


RJMP crc11_execute_Loop


crc11_end:
LSR R17
LSR R17
LSR R17
RET


crc3:
MOV R18, R16
LDI R17, 0
RCALL crc11_execute
OR R17, R18
RET


crc11_check:
```

```
RCALL crc11_execute

RET



crc3_check:

LDI R17, 0x00

RCALL crc11_execute

RET



init:

LDI R16, 0b000 << 5

RCALL crc3;

RCALL transmit

RET



service_readout:

IN R16, Control_Switch

SBRS R16,1

RCALL last_entry

MOV YL, XL

MOV YH, XH



dump:

LD R16, -Y

STS Packet_Out, R16

LDI R16, 0x00

LDI R17, 0x01

CP YL, XL
```

```
CPC YH, XH
BRNE dump
RET


last_entry:
In R16, Control_Switch
SBRS R16,2
RJMP filler
MOV YL, XL
MOV YH, XH
LD R16, -Y
STS Packet_Out, R16


filler:
RET


check_stack:
PUSH R16
PUSH R17
In R16, SPL
In R17, SPH
LDI R18, low(RAMEND) - 4
LDI R19, high(RAMEND)
CP R16, R18
CPC R17,R19
BREQ s0
LDI R18,0
RJMP s1
```

```
s0:

LDI R18, 1

s1:

POP R17

POP R16

RET



transmit:

OUT Packet_OUT, R17

RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;

;start of code from here

power: ; CONFIGURATIONS

Port_Config

XMEM_Config

Pointer_Config



main:                ; Initialization and Push TOS

Stack_Init

rcall INIT

push R17



start:                       ; Master loop (FLow chart layout followed)

RCALL service_readout

IN r16, Control_Switch
```

```
ANDI r16, 0b00000111
CPI r16, 1
brne main


LDI R16, 0x10
STS Ready, R16


LDS R16, Receive
SBRS R16, 3
RJMP start


recieve_on:
LDS R16, Receive
SBRC R16, 3
RJMP Recieve_on


LDI R16, Zeros
STS Ready, R16


IN R21, Packet_In


SBRC R21, 7
RJMP not_command
RJMP tos_data
```

```
not_command:
RCALL check_stack
SBRS R18, 0
pop R22
push R21
RJMP start


tos_data:
POP R23
PUSH R23


SBRC R23, 7
RJMP check11
RJMP check3


check11:
MOV R16, R21
MOV R17, R23
RCALL crc11_check
CPI R17, 0
BREQ state
pop R22
RCALL repeat_request
RJMP start
state:
ANDI R21, 0b11100000
CPI R21, 0b001 << 5
```

```
BRNE main
POP R23
MOV R24, R23
RCALL overwrite_check
ST X+, R24
LDI R16, 0x20
push R16
RCALL crc3
RCALL transmit
RJMP start


repeat_request:
LDI R16, 0x60
RCALL crc3
RCALL transmit
RET


overwrite_check:
LDI R16, 0xEA
LDI R17, 0x10
CP XL, R16
CPC XH, R17
BREQ xmem
RJMP oc1
```

```
xmem:
LDI XL, 0x00
LDI XH, 0x11
RJMP oc2


oc1:
LDI R16, 0xFF
LDI R17, 0xFF
CP XL, R16
CPC XH, R17
BREQ int_mem
RJMP oc2


int_mem:
LDI XL, 0x00
LDI XH, 0x01


oc2:
RET


check3:
MOV R16, R21
RCALL crc3_check
CPI R17,0
BREQ check3_pass
RCALL repeat_request
```

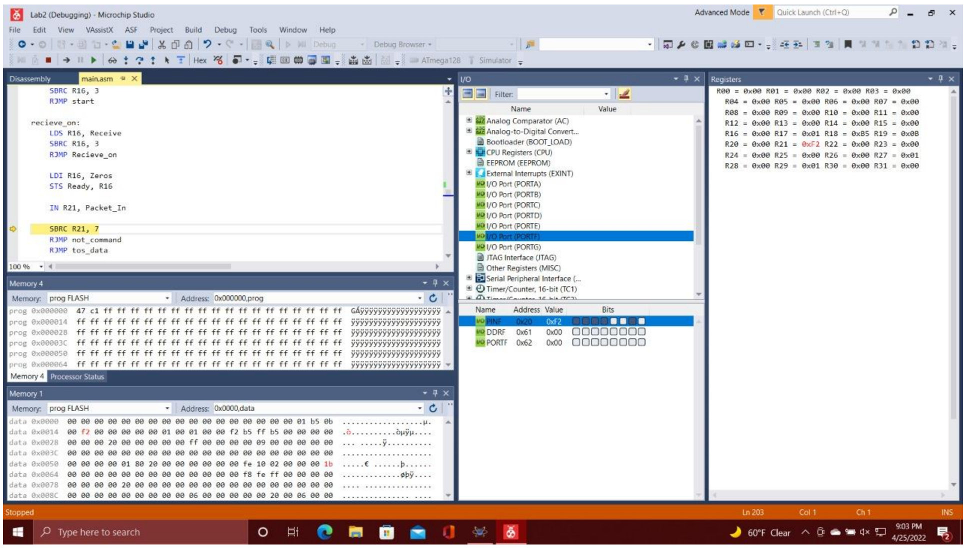
```
RJMP start


check3_pass:
ANDI R21, 0b11100000
CPI R21, 0b010 << 5
BRNE ack_fail
RCALL check_stack
SBRC R18, 0
RJMP start
POP R18
RJMP start


ack_fail:
ANDI R21, 0b11100000
CPI R21, 0b011 << 5
BRNE fail
RCALL check_stack
SBRC R18, 0
RJMP start
POP R17
out Packet_Out, R17
RJMP start


fail:
RJMP start
```

Capturing packet into R16



```
    SBRC R16, 3
    RJMP start

recieve_on:
    LDS R16, Receive
    SBRC R16, 3
    RJMP Recieve_on

    LDI R16, Zeros
    STS Ready, R16

    IN R21, Packet_In

    SBRC R21, 7
    RJMP not_command
    RJMP tos_data
```

Memory 4 — Memory: prog FLASH Address: 0x000000,prog

```
prog 0x000000  47 c1 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  GÁÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x000014  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x000028  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x00003C  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x000050  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x000064  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Memory 1 — Memory: prog FLASH Address: 0x0000,data

```
data 0x0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 b5 0b  .................µ.
data 0x0014  00 f2 00 00 00 00 00 01 00 01 00 00 f2 b5 ff b5 00 00 00 00  .ò.........òµÿµ....
data 0x0028  00 00 00 20 00 00 00 00 00 ff 00 00 00 00 09 00 00 00 00 00  ... .....ÿ..........
data 0x003C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0050  00 00 00 00 01 80 20 00 00 00 00 00 00 fe 10 02 00 00 00 1b  .....€ ......þ......
data 0x0064  00 00 00 00 00 00 00 00 00 00 00 00 00 f8 fe ff 00 00 00 00  .............øþÿ....
data 0x0078  00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .... ...............
data 0x008C  00 00 00 00 00 00 00 00 00 06 00 00 00 00 00 20 00 06 00 00  ............... ....
```

I/O — Filter:

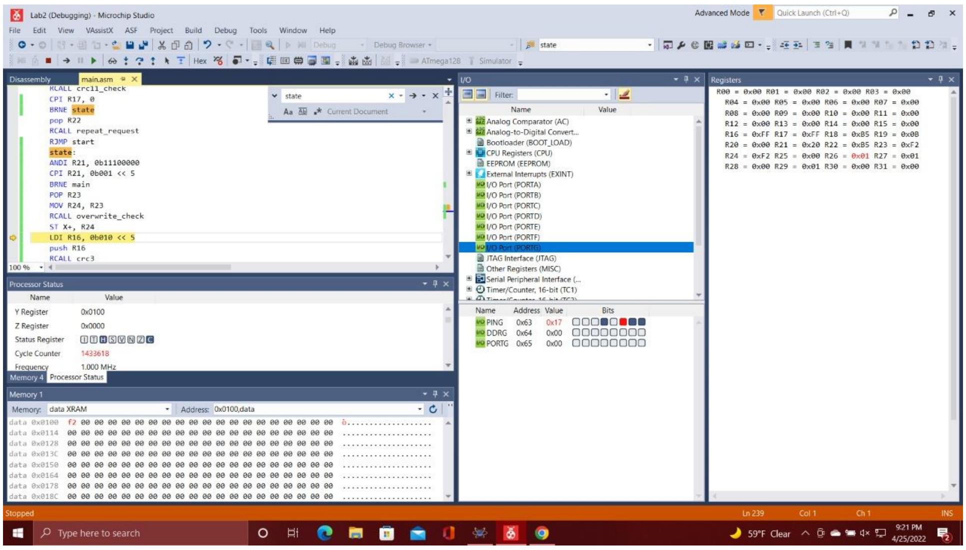
- ⊞ Analog Comparator (AC)
- ⊞ Analog-to-Digital Convert...
- Bootloader (BOOT_LOAD)
- ⊞ CPU Registers (CPU)
- EEPROM (EEPROM)
- ⊞ External Interrupts (EXINT)
- I/O Port (PORTA)
- I/O Port (PORTB)
- I/O Port (PORTC)
- I/O Port (PORTD)
- I/O Port (PORTE)
- I/O Port (PORTF)
- I/O Port (PORTG)
- JTAG Interface (JTAG)
- Other Registers (MISC)
- ⊞ Serial Peripheral Interface (...
- ⊞ Timer/Counter, 16-bit (TC1)

Name	Address	Value	Bits
PINF	0x20	0xF2	■■■■□□■□
DDRF	0x61	0x00	□□□□□□□□
PORTF	0x62	0x00	□□□□□□□□

Registers

```
R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00
  R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00
  R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
  R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00
  R16 = 0x00 R17 = 0x01 R18 = 0xB5 R19 = 0x0B
  R20 = 0x00 R21 = 0xF2 R22 = 0x00 R23 = 0x00
  R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x01
  R28 = 0x00 R29 = 0x01 R30 = 0x00 R31 = 0x00
```

Data Packet is logged into memory as log command was received.



```
    RCALL crc11_check
    CPI R17, 0
    BRNE state
    pop R22
    RCALL repeat_request
    RJMP start
    state:
    ANDI R21, 0b11100000
    CPI R21, 0b001 << 5
    BRNE main
    POP R23
    MOV R24, R23
    RCALL overwrite_check
    ST X+, R24
    LDI R16, 0b010 << 5
    push R16
    RCALL crc3
```

Processor Status

Name	Value
Y Register	0x0100
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1433618
Frequency	1.000 MHz

Memory 1 — Memory: data XRAM Address: 0x0100,data

```
data 0x0100  f2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ò...................
data 0x0114  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x013C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0164  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0178  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x018C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
```

I/O — Filter:

- ⊞ Analog Comparator (AC)
- ⊞ Analog-to-Digital Convert...
- Bootloader (BOOT_LOAD)
- ⊞ CPU Registers (CPU)
- EEPROM (EEPROM)
- ⊞ External Interrupts (EXINT)
- I/O Port (PORTA)
- I/O Port (PORTB)
- I/O Port (PORTC)
- I/O Port (PORTD)
- I/O Port (PORTE)
- I/O Port (PORTF)
- I/O Port (PORTG)
- JTAG Interface (JTAG)
- Other Registers (MISC)
- ⊞ Serial Peripheral Interface (...
- ⊞ Timer/Counter, 16-bit (TC1)

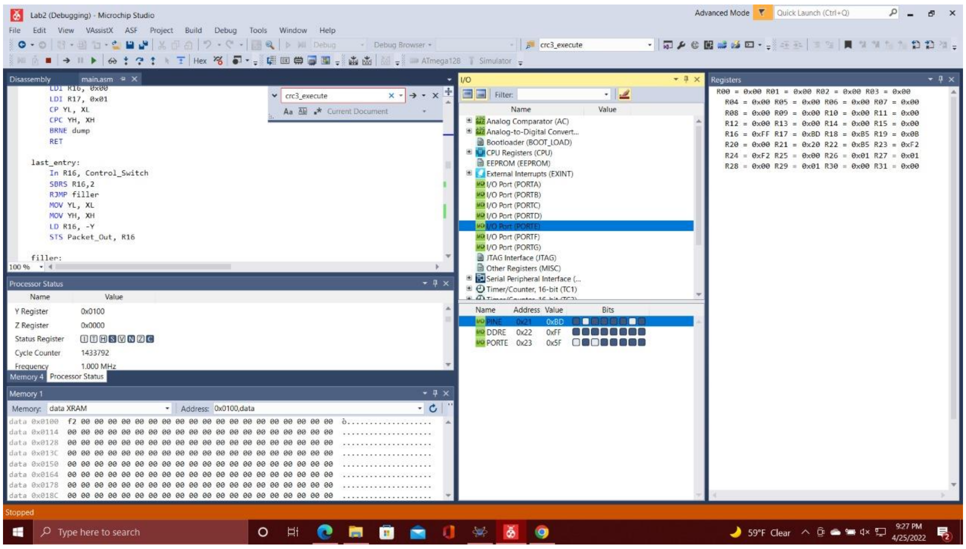
Name	Address	Value	Bits
PING	0x63	0x17	□□□■□■■■
DDRG	0x64	0x00	□□□□□□□□
PORTG	0x65	0x00	□□□□□□□□

Registers

```
R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00
  R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00
  R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
  R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00
  R16 = 0xFF R17 = 0xFF R18 = 0xB5 R19 = 0x0B
  R20 = 0x00 R21 = 0x20 R22 = 0xB5 R23 = 0xF2
  R24 = 0xF2 R25 = 0x00 R26 = 0x01 R27 = 0x01
  R28 = 0x00 R29 = 0x01 R30 = 0x00 R31 = 0x00
```

CRC3 has an issue will fix before demo

After packet was logged, acknowledgement was transmitted to packet out.



```
    LDI R16, 0x00
    LDI R17, 0x01
    CP YL, XL
    CPC YH, XH
    BRNE dump
    RET

last_entry:
    In R16, Control_Switch
    SBRS R16,2
    RJMP filler
    MOV YL, XL
    MOV YH, XH
    LD R16, -Y
    STS Packet_Out, R16

filler:
```

Processor Status

Name	Value
Y Register	0x0100
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1433792
Frequency	1.000 MHz

Memory 1 — Memory: data XRAM — Address: 0x0100,data

```
data 0x0100  f2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ò...................
data 0x0114  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x013C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0164  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0178  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x018C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
```

I/O

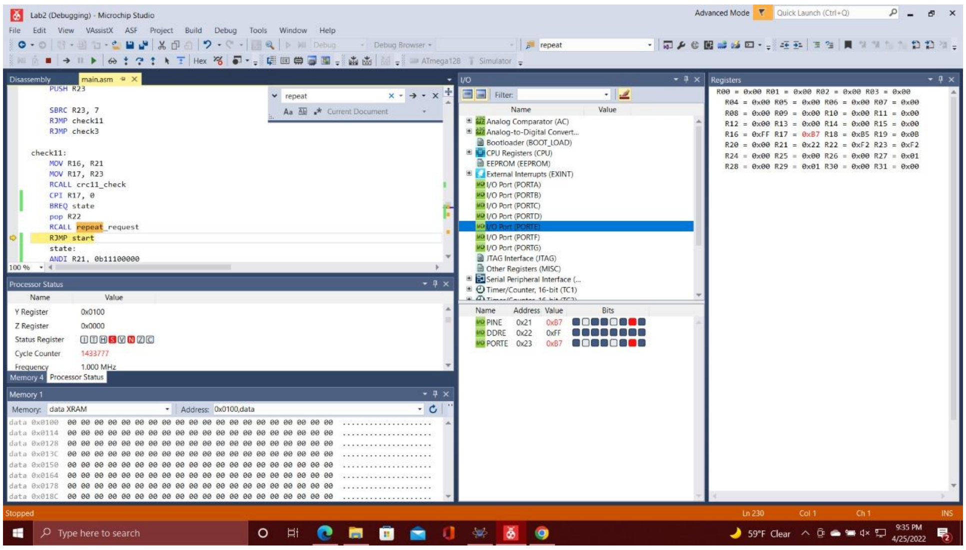
Analog Comparator (AC)
Analog-to-Digital Convert...
Bootloader (BOOT_LOAD)
CPU Registers (CPU)
EEPROM (EEPROM)
External Interrupts (EXINT)
I/O Port (PORTA)
I/O Port (PORTB)
I/O Port (PORTC)
I/O Port (PORTD)
I/O Port (PORTE)
I/O Port (PORTF)
I/O Port (PORTG)
JTAG Interface (JTAG)
Other Registers (MISC)
Serial Peripheral Interface (...
Timer/Counter, 16-bit (TC1)

Name	Address	Value
PINE	0x21	0xBD
DDRE	0x22	0xFF
PORTE	0x23	0x5F

Registers

```
R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00
 R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00
 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
 R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00
 R16 = 0xFF R17 = 0xBD R18 = 0xB5 R19 = 0x0B
 R20 = 0x00 R21 = 0x20 R22 = 0xB5 R23 = 0xF2
 R24 = 0xF2 R25 = 0x00 R26 = 0x01 R27 = 0x01
 R28 = 0x00 R29 = 0x01 R30 = 0x00 R31 = 0x00
```

Repeat request successful since log command was incorrect



```
    PUSH R23

    SBRC R23, 7
    RJMP check11
    RJMP check3

check11:
    MOV R16, R21
    MOV R17, R23
    RCALL crc11_check
    CPI R17, 0
    BREQ state
    pop R22
    RCALL repeat_request
    RJMP start
    state:
    ANDI R21, 0b11100000
```

Processor Status

Name	Value
Y Register	0x0100
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1433777
Frequency	1.000 MHz

Memory 1 — Memory: data XRAM — Address: 0x0100,data

```
data 0x0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0114  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x013C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0164  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x0178  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
data 0x018C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
```

I/O

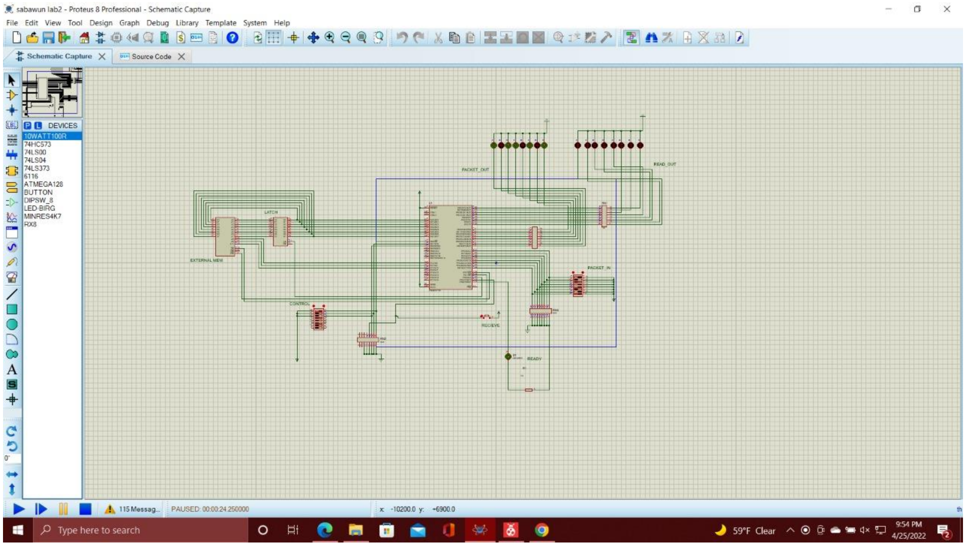
Analog Comparator (AC)
Analog-to-Digital Convert...
Bootloader (BOOT_LOAD)
CPU Registers (CPU)
EEPROM (EEPROM)
External Interrupts (EXINT)
I/O Port (PORTA)
I/O Port (PORTB)
I/O Port (PORTC)
I/O Port (PORTD)
I/O Port (PORTE)
I/O Port (PORTF)
I/O Port (PORTG)
JTAG Interface (JTAG)
Other Registers (MISC)
Serial Peripheral Interface (...
Timer/Counter, 16-bit (TC1)

Name	Address	Value
PINE	0x21	0xB7
DDRE	0x22	0xFF
PORTE	0x23	0xB7

Registers

```
R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00
 R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00
 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
 R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00
 R16 = 0xFF R17 = 0xB7 R18 = 0xB5 R19 = 0x0B
 R20 = 0x00 R21 = 0x22 R22 = 0xF2 R23 = 0xF2
 R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x01
 R28 = 0x00 R29 = 0x01 R30 = 0x00 R31 = 0x00
```

Proteus (schematic/infrastruture wiring) lights/switches all working, only the crc3 logic causes incorrect lights to turn on due to the values. Will try and fix this till the demo


sabawun lab2 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Source Code
P L DEVICES
10WATT100R
74HC573
74LS00
74LS04
74LS373
6116
ATMEGA128
BUTTON
DIPSW_8
LED-BIRG
MINRES4K7
RX8
PACKET_OUT
READ_OUT
LATCH
EXTERNAL MEM
PACKET_IN
CONTROL
RECIEVE
READY
0°
115 Messag...
PAUSED: 00:00:24.250000
x: -10200.0 y: -6900.0
Type here to search
59°F Clear
9:54 PM
4/25/2022